

**Implementation of Some Finite Difference Methods for the  
Pricing of Derivatives using C++ Programming.**

By  
Ebenezer Ampadu  
A Masters Project  
Submitted to the Faculty  
Of  
WORCESTER POLYTECHNIC INSTITUTE  
In partial fulfillment of the requirements for the  
Degree of Professional Master of Science  
In  
Financial Mathematics  
May 2007

APPROVED:

PROFESSOR DOMOKOS VERMES

**TABLE OF CONTENTS**

Table Of Contents	2
Acknowledgements	3
<b>ABSTRACT</b>	4
<b>IMPLEMENTATION OF SOME FINITE DIFFERENCE METHODS FOR THE PRICING OF OPTIONS ON DERIVATIVES USING C++</b>	5
1.INTRODUCTION	5
1.1ROADMAP	6
<b>PRICING OF FINANCIAL DERIVATIVES</b>	7
2.1OVERVIEW OF STOCK OPTIONS	7
2.1.1BASIC DEFINITIONS	7
2.2PRICING OF STOCK OPTIONS	7
2.2.1 DISCOUNTED EXPECTATION(MARTINGALE) METHOD	8
2.2.2BLACK&SCHOLES PARTIAL DIFFERENTIAL EQUATION	9
<b>3.OVERVIEW OF NUMERICAL SOLUTION METHODS</b>	10
3.1MONTE CARLO (MAY BE USED FOR MARTINGALE METHOD)	10
3.2 FINITE DIFFERENCE METHODS FOR DERIVATIVES PRICING	11
3.2.1A BRIEF HISTORY OF THE FINITE DIFFERENCE METHOD	11
3.2.2DEFINITION AND DESCRIPTION OF THE FINITE DIFFERENCE METHOD	12
3.2.3.THE EXPLICIT FINITE DIFFERENCE METHOD	15
3.2.4.THE IMPLICIT FINITE DIFFERENCE METHOD	16
3.2.5.THE CRANK-NICOLSON FINITE DIFFERENCE METHOD	18
<b>4.COMPUTATIONAL IMPLEMENTATION/METHODOLOGY</b>	19
<b>5.CONCLUSION</b>	20
<b>6.REFERENCES</b>	21

### **ACKNOWLEDGEMENT**

This simple course project is dedicated to Professor Vermes and all lecturers of the mathematical sciences department.

## **ABSTRACT**

In this project, European Call and Put options, and also American Call and Put options have been priced by some finite difference methods using the C++ programming language. The report describes the following: The theory behind the pricing of options, some pricing methods, and how some finite difference pricing methods have been implemented in C++.

## **IMPLEMENTATION OF SOME FINITE DIFFERENCE METHODS FOR THE PRICING OF DERIVATIVES USING C++ PROGRAMMING.**

### **1.INTRODUCTION**

This project is about the pricing of options by some finite difference methods in C++. European call and put options and also American call and put options will be priced by the Explicit and Implicit finite difference methods in this project. Even though C++ is being used to price options in the financial field, it is not known who has done this project in this manner before. This project provides a direct means to price the options. The user need not have a knowledge of C++ in order to make use of the software that will be attached. This may be the difference between this project and existing ones where a user needs to have a working knowledge of C++ in addition to knowing how to price derivatives. The work may be extended to price derivatives using the Crank-Nicolson method. The goals of this project are the following: Compute Asset prices at maturity for call and put options for both European and American type options, compute option prices at maturity, and last but not the least to back-track into the mesh to compute option prices today. These goals were achieved using abstract factory console programs written in C++ for each option type(call or put) for both European and American type options. The results will be displayed in arrays and matrix / mesh forms. The client-server interaction is self-directional. The simplicity of the source codes will enable clients to use it without having to be trained by professional programmers. Of course, the impact of this is the elimination of training time and the complexities of programming. In view of this, the source codes/software that will be attached may compete with some of the existing software on the market. The outline presented here will provide the reader with a bit of training on derivatives pricing if the individual is not already familiar with pricing methods, or it may provide an already experienced pricer with information or insight into derivatives pricing theory. A roadmap will be provided shortly to serve as a guide to the reader.

### **1.1 Roadmap**

**Section 2.** will be the body of this project. It will give the reader an overview of the basic definitions of stock options and the theory behind the pricing of stock options. This section will also give examples of the methods that apply these theories.

**Section 3.** will concisely describe the pricing methods that have been outlined above. It will also describe the object of this project, namely pricing of options by the finite difference methods.

**Section 4.** will be about the Computational Implementation of the finite difference methods. There, the C++ abstract factory code that implements these methods will be attached.

**Section 5** will conclude this project.

**Section 6.** will cite the references that will be used.

## 2. PRICING OF FINANCIAL DERIVATIVES

A full course on C++ design and implementation is required to do this project. The user only needs to understand derivatives pricing using the finite difference methods well in order to make use of the code that will be attached.

### 2.1 Overview of Stock options

This section will give the reader an overview of stock option definitions, the theories behind option pricing and also how these theories will be incorporated into the real applications that are used to price the options.

#### 2.1.1 Basic Definitions

Stock options are financial derivatives which gives the holder the right to buy or sell stock.

Options are traded both on exchanges and in the over-the-counter market. There are two basic types of options: A call and a put option. A call option gives the holder the right to buy the underlying asset by a certain date for a certain price. A put option gives the holder the right to sell the underlying asset by a certain date for a certain price. The price in contrast is known as the exercise price or strike price; the date in the contract is known as the expiration date or maturity. American options can be exercised at any time up to the expiration date. European options can be exercised only on the expiration date itself. Thus the difference between American and European option is the American option's ability to exercise early. Most of the options that are traded on exchanges are American. In the exchange-traded equity options market, a **contract** is usually an agreement to buy or sell 100 shares. European options are generally easier to analyze than American options, and some of the properties of an American option are frequently deduced from those of its European counterpart.

It should be emphasized that an option gives the holder just the right. The holder does not necessarily have to exercise that right. This is what distinguishes options from forwards and futures contracts, where the holder is obligated to buy or sell the underlying asset. Note that whereas it costs nothing to enter into a forward or futures contract, there is a cost to acquiring an option.

### 2.2 Pricing of Stock options

Stock option prices are assumed to follow a Markov process. This means that they follow a stochastic process where only the present value of a variable is relevant for predicting the future. The past history of the variable and the way that the present has emerged from the past are irrelevant. The Markov property of stock option prices is consistent with the weak form of market efficiency. This states that the present price of a stock option impounds all the information contained in a record of past prices. It is competition in the market price that tends to ensure that weak-form market efficiency holds.

There are two major theories behind the pricing of financial derivatives and these major theories can be incorporated into applications. The processes will be explained in the following:

### 2.2.1 Discounted Expectation(MARTINGALE) Method

A martingale is a zero-drift stochastic process. A variable  $\theta$  follows a martingale if its process has the form  $d\theta = \sigma dz$ , where  $dz$  is a Wiener process. The variable  $\sigma$  may itself be stochastic. It can depend on  $\theta$  and other stochastic variables. A martingale has the convenient property that its expected value at any future time is equal to its value today. This means that

$E(\theta_T) = \theta_0$  where  $\theta_0$  and  $\theta_T$  denote the values of  $\theta$  at times zero and  $T$ , respectively. To understand this result, it must be noted that over a very small time interval the change in  $\theta$  is normally distributed with zero mean. The expected change in  $\theta$  over any very small time interval is therefore zero. The change in  $\theta$  between time zero and time  $T$  is the sum of its changes over many small time intervals. It follows that the expected change in  $\theta$  between time zero and time  $T$  must also be zero.

Suppose that  $f$  and  $g$  are the prices of traded securities dependent on a single source of uncertainty. It is assumed that the securities provide no income during the time period under consideration. Define  $\Phi = f/g$ . The variable  $\Phi$  is the relative price of  $f$  with respect to  $g$ . One may think of  $\Phi$  as measuring the price of  $f$  in units of  $g$  rather than in dollars. The security price  $g$  is called the numeraire. The equivalent martingale measure result shows that, when there are no arbitrage opportunities,  $\Phi$  is a martingale for some choice of the market price of risk. What is more, for a given numeraire security  $g$ , the same choice of the market price of risk makes  $\Phi$  a martingale for all securities  $f$ . This choice of the market price of risk is the volatility of  $g$ . In other words, when the market price of risk is set equal to the volatility of  $g$ , the ratio  $f/g$  is a martingale for all security prices  $f$ . To prove this result, one supposes that the volatilities of  $f$  and  $g$  are  $\sigma_f$  and  $\sigma_g$ . From the following equation:  $df = (r + \lambda\sigma)fdt + \sigma_f dz$ , in a world where the market price of risk is  $\sigma_g$ , an individual obtains

$$df = (r + \sigma_g \sigma_f)fdt + \sigma_f dz$$

$$dg = (r + \sigma_g^2)gdt + \sigma_g dz$$

Using Ito's lemma gives

$$d \ln f = (r + \sigma_g \sigma_f - \sigma_f^2/2)dt + \sigma_f dz$$

$$d \ln g = (r + \sigma_g^2/2)dt + \sigma_g dz$$

so that

$$d(\ln f - \ln g) = (\sigma_g \sigma_f - \sigma_f^2/2 - \sigma_g^2/2)dt + (\sigma_f - \sigma_g)dz$$

$$d(\ln f/g) = -((\sigma_f - \sigma_g)^2/2)dt + (\sigma_f - \sigma_g)dz.$$

Using Ito's lemma to determine the process for  $f/g$  from the process for that  $\ln(f/g)$ , one obtains  $d(f/g) = ((\sigma_f - \sigma_g)/g)dz$  showing that  $f/g$  is a martingale.

This provides the required result. Make reference to a world where the market price of risk is  $\sigma_g$  is a world that is forward risk neutral with respect to  $g$ . Because  $f/g$  is a martingale in a world that is forward risk neutral with respect to  $g$ , it follows that

$f_0/g_0 = E_g(f_T/g_T)$  or  $f_0 = g_0 E_g(f_T/g_T)$  where  $E_g$  denotes the expected value in a world that is forward risk neutral with respect to  $g$ . [1]



One method which can apply this theory is the Monte Carlo Simulation Method for financial derivatives.

### 2.2.2 Black&Scholes Partial Differential Equation

The Black-Scholes-Merton differential equation is an equation that must be satisfied by the price of any derivative dependent on a non-dividend paying stock. The process involves setting up a risk-less portfolio consisting of a position in the derivative and a position in the stock. In the absence of arbitrage opportunities, the return from the portfolio must be the risk-free interest rate,  $r$ . The reason a risk-less portfolio can be set up is that the stock price and the derivative price are both affected by the same underlying source of uncertainty :stock price movements. In any short period of time, the price of the derivative is perfectly correlated with the price of the underlying stock. When an appropriate portfolio of the stock and the derivative is established, the gain or loss from the stock position always offsets the gain or loss from the derivative position so that the overall value of the portfolio at the end of the short period of time is known with certainty.[1]

Suppose, for example ,that at a particular point in time the relationship between a small change in the stock price,  $\delta S$ , and the resultant small change in the price of a European call option,  $\delta c$ , is given by

$$\delta c = 0.4 \delta S.$$

This means that the slope of the line representing the relationship between  $c$  and  $S$  is 0.4. The Black and Scholes partial differential equation is given by

$$-\delta C/\delta t = 1/2 * (S^2 \sigma^2) * (\delta^2 C/\delta S^2) + (S * (r - \delta)) (\delta C/\delta S) - rC \dots (1)$$

Methods which makes use of the Black-Scholes PDE are the finite difference methods.

### 3.OVERVIEW OF NUMERICAL SOLUTION METHODS

This section will provide the reader with a more concise description of the application of the theories that have been described above. The theories will be incorporated into two methods namely :The Monte Carlo Simulation Method for financial Derivatives and the Finite Difference Methods.

#### 3.1Monte Carlo (May be used for Martingale method)

A Monte Carlo simulation of a stochastic process is a procedure for sampling random outcomes for the process. However when the number of dimensions (or degrees of freedom) in the problem is large, PDE' s and numerical integrals become intractable, and in these cases **Monte Carlo methods** often give better results.

The Monte Carlo method for derivative pricing makes use of risk-neutral valuation. The expected payoff in a risk-neutral world is calculated using a sampling procedure. It is then discounted at the risk-free interest rate.

Consider a derivative dependent on a single market variable S that provides a payoff at time T. Assuming that interest rates are constant,one can value the derivative as follows:

- 1.Sample a random path for S in a risk-neutral world.
- 2.Calculate the payoff from the derivative.
- 3.Repeat steps 1.&2. to get many sample values of the payoff from the derivative in a risk-neutral world.
- 4.Calculate the mean of the sample payoffs t get an estimate of the expected payoff in a risk-neutral world.
5. Discount the expected payoff at the risk-free rae to get an estimate of the value of the derivative.

Suppose that the process followed by the underlying market variable in a risk-neutral world is

$dS = \mu S dt + \sigma S dz \dots (y)$  where dz is a Weiner process,  $\mu$  is the expected return in a risk-neutral world, and  $\sigma$  is the volatility. Note that  $\mu = r - r_f$ . To simulate the path followed by S,one divides the life of the derivative into N short intervals of length  $\delta t$  and approximate the above equation as

$S(t+dt) - S(t) = \mu S(t) \delta t + \sigma S(t) \epsilon(\sqrt{\delta t}) \dots (x)$  where S(t) denotes the value of S at time t, and  $\epsilon$  is a random sample from a normal distribution with mean zero and standard deviation 1.0.This enables the value of S at time  $\delta t$  to be calculated from the initial value of S,the value at time  $2\delta t$  to be calculated from the value at time  $\delta t$ , and so on.

One simulation trial involves constructing a complete path for  $S$  using  $N$  random samples from a normal distribution. In practice, it is usually more accurate to simulate  $\ln S$  rather than  $S$ . From Ito's lemma the process followed by  $\ln S$  is

$d\ln S = (\mu - (\sigma^2/2))dt + \sigma dz$  ... (z) so that

$\ln S(t+\delta t) - \ln S(t) = (\mu - \sigma^2/2)\delta t + \sigma \varepsilon(\sqrt{\delta t})$  or equivalently

$S(t+\delta t) = S(t)\exp[(\mu - \sigma^2/2)\delta t + \sigma \varepsilon(\sqrt{\delta t})]$ .

This equation is used to construct a path for  $S$  in a similar way to equation (x).

Whereas equation (x) is true only in the limit as  $\delta t$  tends to zero, equation (z) is exactly true for all  $\delta t$ .

The advantage of Monte Carlo simulation is that it can be used when the payoff depends on the path followed by the underlying variable  $S$  as well as when it depends only on the final value of  $S$ . Payoffs can occur at several times during the life of the derivative rather than all at the end. Any stochastic process for  $S$  can be accommodated. The procedure can also be extended to accommodate situations where the payoff from the derivative depends on several underlying market variables.

In the particular case where a derivative provides a payoff at time  $T$  dependent only on the value of  $S$  at that time, it is not necessary to sample a whole path for  $S$ . Instead we can jump straight from the value of  $S$  at time zero to its value at time  $T$ . When the process in equation (y) is assumed,

$S(T) = S(0)\exp[(\mu - \sigma^2/2)T + \sigma \varepsilon(\sqrt{T})]$ .

Instead of implementing Monte Carlo simulation by randomly sampling from the stochastic process for an underlying variable, one can sample paths for the underlying variable using a binomial tree.

The drawbacks of Monte Carlo simulation are that it is computationally very time-consuming and cannot easily handle situations where there are early exercise opportunities.[1] This is where the finite difference methods come in handy.

Monte Carlo simulation may also be used to compute VaR for portfolios containing securities with non-linear returns (e.g. options) since the computational effort required is non-trivial. Note that for portfolios without these complicated securities, such as a portfolio of stocks, the variance-covariance method is perfectly suitable and should probably be used instead. [2]

### 3.2 Finite Difference Methods for Derivatives Pricing

The implementation of these methods is the focus of this project.

This section will give a brief history of the finite difference methods, and concise descriptions of the different versions of the finite difference methods.

#### 3.2.1A Brief History of the finite Difference Method

The finite difference method was invented by a Chinese scientist named Feng Kang in the late 1950's. He proposed the finite difference method as a systematic numerical method for solving partial differential equations that are applied to the computations of dam constructions. It is speculated that the same method was also independently invented in

the West, named in the West the finite element method. It is now considered that the invention of the finite difference method is a milestone of computational mathematics.[2]

### 3.2.2 Definition and Description of the Finite Difference Methods

In a simplistic way, a finite difference is defined as a mathematical expression of the form  $f(x+b)-f(x+a)$ . If a finite difference is divided by  $(b-a)$ , one gets an expression similar to a differential quotient, except that it uses finite quantities instead of infinitesimal ones. The approximation of derivatives by finite differences plays a central role in finite difference methods for the numerical solution of partial differential equations.

Generally, the finite difference method is used for finding approximate solutions of partial differential equations (PDE's) as well as for the solutions of integral equations such as the heat transfer equation. The solution approach is based either on eliminating the differential equation completely (steady state problems), or rendering the PDE into an equivalent ordinary differential equation, which is then solved using standard techniques such as finite differences.

In solving partial differential equations, the primary challenge is to create an equation which approximates the equation to be studied, but which is numerically stable, meaning that errors in the input data and intermediate calculations do not accumulate and cause the resulting output to be meaningless. There are many ways of doing this, all with advantages and disadvantages. The Finite difference Method is a good choice for solving partial differential equations over complex domains. Examples are the methods' applications to automobile transmission development and in oil pipelines, where the domain changes. The method is also applicable to a solid state reaction with a moving boundary, or when the desired precision varies over the entire domain. For instance, in simulating the weather pattern on Earth, it is more important to have accurate predictions over land than over the wide-open sea, a demand that is achievable using the finite difference method.

The finite difference method applies three main difference schemes: The forward difference, the central difference, and the backward difference.

A forward difference is an expression of the form  $\Delta[f](x)=f(x+h)-f(x)$ . Depending on the application, the spacing  $h$  is held constant, or the limit  $h \rightarrow 0$  is taken.

A backward difference arises when  $h$  is replaced by  $-h$ :  $\nabla[f](x)=f(x)-f(x-h)$ . Finally, the central difference is given by

$$\Delta[f](x) = f(x + h/ 2) - f(x -h/ 2).$$

**How finite differences relate with derivatives**

The derivative of a function  $f$  at a point  $x$  is defined by the limit

$f'(x) = \lim_{h \rightarrow 0} [f(x+h) - f(x)]/h$ . If  $h$  has a fixed (non-zero) value, instead of approaching zero, then the right-hand side is  $[f(x+h) - f(x)]/h = \Delta[f](x)/h$ . Hence, the forward difference divided by  $h$  approximates the derivative when  $h$  is small. The error in this approximation can be derived from Taylor's theorem. Assuming that  $f$  is continuously differentiable, the error is  $[\Delta[f](x)/h] - f'(x) = O(h)$  ( $h \rightarrow 0$ ).

The same formula holds for the backward difference:

$$\{\nabla[f](x)\}/h - f'(x) = O(h).$$

However, the central difference yields a more accurate approximation. Its error is proportional to square of the spacing (if  $f$  is twice continuously differentiable):

$$[\Delta[f](x)]/h - f'(x) = O(h^2).$$

### Calculus of finite differences

The forward difference can be considered as a difference operator, which maps the function  $f$  to  $\Delta f$ . Taylor's theorem can be expressed by the formula

$$\Delta = hD + h^2 D^2/2 + h^3 D^3/3! + \dots = e^{hD} - 1,$$

where  $D$  denotes the derivative operator, mapping  $f$  to its derivative  $f'$ . Formally inverting the exponential suggests that  $hD = \log(1 + \Delta) = \Delta - \Delta^2/2 + \Delta^3/3 + \dots$ ,

This formula holds in the sense that both operators give the same result when applied to a polynomial. Even for analytic functions, the series on the right is not guaranteed to converge; it may be an asymptotic series. However, it can be used to obtain more accurate approximations for the derivative. For instance, retaining the first two terms of the series yields

$$f'(x) \approx \{\Delta[f](x) - [\Delta^2[f](x)]/2\}/h = \{f(x+2h) - 4f(x+h) + 3f(x)\}/2h.$$

The error in this approximation is of the order  $h^2$ . The analogous formulas for the backward and central difference operators are  $hD = -\log(1 - \nabla)$  and  $hD = \operatorname{arcsinh}(\delta)$ .

In an analogous way, one can obtain finite difference approximations to higher order derivatives and differential operators. For example, by using the above central difference formula with step  $h/2$  for  $f'(x + h/2)$  and  $f'(x - h/2)$  and then applying a central difference formula for the derivative of  $f'$  at  $x$ , one obtains the central difference approximation of the second derivative of  $f$ :

$$f''(x) = \delta^2[f](x)/h^2 = [f(x+h) - 2f(x) + f(x-h)]/h^2.$$

## More applications

The most common applications of the finite difference methods are in computational science and engineering disciplines, such as thermal engineering, fluid mechanics, etc.

In recent years, the finite difference schemes have been constructed and tested for a wide range of nonlinear dynamical systems in the following areas:

Singular boundary value problems expressed in spherical or cylindrical coordinates.

A generalized Nagumo reaction-diffusion model.

Equations modeling stellar structure

The dynamics of HIV transmission

Modified linear heat/diffusion transport problems.

Finite difference schemes are enjoying a growing applicability as the practical users of numerical techniques for differential equations become aware of the advantages and power of these methods. However, there is the view held by some individuals that the finite difference methods depends on using known solution of the differential or by ad hoc experimentation. Recent studies shows that this information is false in that a lot has been accomplished using this method. An essential issue is that each differential equation has to be considered a unique mathematical structure and consequently must be modeled in a unique manner. [2]

There are three main kinds of finite difference methods:

1)The Explicit finite difference Method

2)The Implicit finite difference Method

3)The Crank-Nicolson finite difference Method

In this project, the finite difference methods has been applied to the Black- Scholes PDE by replacing the partial differentials with finite differences. The Black-Scholes PDE has been given by equation (1) above.

This PDE governs the price of all European and American options whose pay-offs depend on a single asset which follows the Geometric Brownian Motion. In terms of  $x=\ln(S)$ , the above equation becomes  $-\delta C/\delta t=1/2*(\sigma^2)*(\delta^2 C/\delta x^2)+v*\delta C/\delta x-rC\dots(2)$ .

The foregoing equation is a PDE with constant coefficients ,i.e. it does not depend on  $x$  or  $t$  which makes the application of finite difference methods much easier. When implementing finite difference methods ,one should imagine time and space divided up into discrete intervals ( $\Delta t$  and  $\Delta x$ ), and this is called the finite difference grid or lattice.

Now, the major kinds of finite difference methods will be discussed.

### 3.2.3. The Explicit Finite Difference Method

The explicit finite difference method is equivalent to a trinomial tree approach.

This is achieved by approximating equation (2) using a forward difference for  $\delta C/\delta t$  and central differences for  $\delta^2 C/\delta x^2$  and  $\delta C/\delta x$ . This implies that in terms of the grid

$-C_{i+1,j} - C_{i,j} / \Delta t = 1/2 * \sigma^2 C_{i,j,j+1} - 2C_{i+1,j} + C_{i+1,j-1} / \Delta x^2 + v C_{i+1,j+1} - C_{i+1,j-1} / 2 \Delta x - r C_{i+1,j}$   
 ... (3) the above is obtained and it can be re-written as

$$C_{i,j} = P_u C_{i+1,j+1} + P_m C_{i+1,j} + P_d C_{i+1,j-1} \dots (4)$$

$$P_u = \Delta t (\sigma^2 / 2 \Delta x^2 + v / 2 \Delta x) \dots (4a)$$

$$P_m = 1 - \Delta t (\sigma^2 / \Delta x^2) - r \Delta t \dots (4b)$$

$$P_d = \Delta t (\sigma^2 / 2 \Delta x^2 - v / 2 \Delta x) \dots (4c)$$

Equation (4) is equivalent to taking discounted expectations. This can be seen by taking a slightly different approximation to the PDE in which the last term in equation (4) is approximated by the value at node (i,j) rather than (i+1,j)

$$C_{i+1,j} - C_{i,j} / \Delta t = 1/2 * \sigma^2 C_{i,j,j+1} - 2C_{i+1,j} + C_{i+1,j-1} / \Delta x^2 + v C_{i+1,j+1} - C_{i+1,j-1} / 2 \Delta x - r C_{i,j} \dots (5)$$

$$\text{and this can be rewritten as } C_{i,j} = 1 / (1 + r \Delta t) * (P_u C_{i+1,j+1} + P_m C_{i+1,j} + P_d C_{i+1,j-1}) \dots (6)$$

$$P_u = \Delta t (\sigma^2 / 2 \Delta x^2 + v / 2 \Delta x) \dots (6a)$$

$$P_m = 1 - \Delta t (\sigma^2 / \Delta x^2) \dots (6b)$$

$$P_d = 1/2 * \Delta t (\sigma^2 / \Delta x^2 - v / \Delta x) \dots (6c);$$

where  $1/(1+r\Delta t)$  is an approximation of  $1/(\exp(r\Delta t))$ . Therefore, the explicit finite difference method is equivalent to approximating the diffusion process by a discrete trinomial process. It can be shown that the variance of the discrete process is a downward-biased approximation of the continuous geometric brownian motion process with an upper bound of  $\sigma^4$  to the bias but that is outside the scope of this project. [3]

### 3.2.4. The Implicit Finite Difference Method

Supposing now that the transformed Black-Scholes PDE is approximated by replacing the space derivatives with central differences at time step  $i$  rather than at  $i+1$ . The following result is obtained:

$$-C_{i+1,j+1} - C_{i,j} / \Delta t = 1/2 * \sigma^2 * C_{i,j+1} - 2C_{i,j} + C_{i,j-1} / \Delta x^2 + v C_{i,j+1} - C_{i,j+1} - C_{i,j-1} / 2 \Delta x - r C_{i,j} \dots (7)$$

and this can be re-written as

$$P_u C_{i,j+1} + P_m C_{i,j} + P_d C_{i,j-1} = C_{i+1,j} \dots (8)$$

$$P_u = -1/2 * \Delta t (\sigma^2 / \Delta x^2 + v / \Delta x) \dots (8a)$$

$$P_m = 1 + \Delta t * \sigma^2 / \Delta x^2 + r \Delta t \dots (8b)$$

$$P_d = -1/2 \Delta t (\sigma^2 / \Delta x^2 - v / \Delta x) \dots (8c)$$

Each equation (8) for  $j = -N_j + 1, \dots, N_j - 1$  cannot be solved individually for the option values at time step 'i' as they could for the explicit finite difference method. Instead, they are considered together with the boundary conditions,

$$C_{i,N_j} - C_{i,N_j-1} = \lambda_u \dots (9)$$

$$C_{i,-N_j+1} - C_{i,-N_j} = \lambda_L \dots (10)$$

to be a system of  $2N_j + 1$  linear equations which implicitly determine the  $2N_j + 1$  option values at time step  $i$ . The boundary condition parameters  $\lambda_u$  and  $\lambda_L$  are determined by the type of option being valued, for example for a call we have

$$\lambda_u = S_{i,N_j} - S_{i,N_j-1} \dots (11)$$

$$\lambda_L = 0 \dots (12)$$

This set of equations has a special structure which is called tri-diagonal. The matrix is zero everywhere except on the main diagonal, and lower and upper diagonals. Each equation has two variables in common with the equation above and below.

This tri-diagonal matrix equation can be solved very efficiently. The diagonals are placed in three separate vectors and the process will be displayed in the code that will be attached.

But for now, beginning with the boundary condition equation  $j = -N_j$ , this equation is rearranged to obtain

$$C_{i,-N_j} = C_{i,-N_j+1} - \lambda_L \dots (13)$$

This is then substituted into the equation above ( $j = -N_j + 1$ ) to obtain

$$P_u C_{i,-N_j+2} + P_m C_{i,-N_j+1} + P_d (C_{i,-N_j+1} - \lambda_L) = C_{i+1,-N_j+1} \dots (14)$$

$$\text{Which can be re-written as } P_u C_{i,-N_j+2} + P^* m C_{i,-N_j+1} = p^* \dots (15)$$

Where

$$P^* m = P_m + P_d \dots (15a) \text{ and } P^* = C_{i+1,-N_j+1} + P_d \lambda_L \dots (15b)$$

Therefore the original equation for  $j = -N_j + 1$  with three unknowns has become equation (15) with only two unknowns. Equation (15) can be rearranged to obtain



$$C_{i,-N_j+1} = P' - P_u C_{i,-N_j+2} / P' m \dots (16)$$

Which can be substituted into the equation for  $j = -N_j + 2$  to obtain

$$P_u C_{i,-N_j+3} + P' m C_{i,-N_j+2} = P' \dots (17)$$

Where

$$P' m = P_m - (P_u / P' m_{,-N_j+1}) P_d \dots (18)$$

$$P' = C_{i+1,-N_j+2} - (P'_{-N_j+1} / P' m_{,-N_j+1}) P_d \dots (19)$$

and subscripts have been added to the p's to indicate that they apply to the equation for  $j = -N_j + 1$ . This process of substitution can be repeated all the way up to  $j = -N_j - 1$  where the following equation is obtained:

$$P_u C_{i,N_j} + P' m C_{i,N_j-1} = p' \dots (20)$$

Now using equation(16) and the boundary condition equation for  $j = N_j$

$$C_{i,N_j} - C_{i,N_j-1} = \lambda u$$

One can solve for both  $C_{i,N_j}$  and  $C_{i,N_j-1}$ . Using the next equation down for  $j = N_j - 2$  and  $C_{i,N_j-1}$ ;  $C_{i,N_j-2}$  is obtained. This process of substitution can be repeated all the way down to  $j = -N_j$ , at which point the tri-diagonal system of equations (the matrix described above) has been solved. [3]

### 3.2.5. The Crank-Nicolson Finite Difference Method

The Crank-Nicolson method is a refinement of the implicit finite difference method. It is a so-called fully centred method because it replaces the space and time derivatives with finite differences centred at an imaginary time step at  $(i+1/2)$ . If this is done, the following finite difference equation is obtained:

$$(-C_{i+1,j} - C_{i,j})/\Delta t = 1/2 * \sigma^2 * \{[(C_{i+1,j+1} - 2C_{i+1,j} + C_{i+1,j-1}) + (C_{i,j+1} - 2C_{i,j} + C_{i,j-1})]/2\Delta x^2\} + v\{[(C_{i+1,j+1} - C_{i+1,j-1}) + (C_{i,j+1} - C_{i,j-1})]/4\Delta x\} - r[(C_{i+1,j} + C_{i,j})/2] \dots (21)$$

and this can be rewritten as

$$P_u C_{i,j+1} + P_m C_{i,j} + P_d C_{i,j-1} = -P_u C_{i+1,j+1} - (P_m - 2) C_{i+1,j} - P_d C_{i+1,j-1} \dots (22)$$

$$P_u = -1/4 \Delta t ([\sigma^2 / \Delta x^2] + [v/\Delta x]) \dots (22a)$$

$$P_m = 1 + \Delta t \sigma^2 / 2 \Delta x^2 + r \Delta t / 2 \dots (22b)$$

$$P_d = -1/4 \Delta t ([\sigma^2 / \Delta x^2] - [v/\Delta x]) \dots (22c)$$

The right-hand side of equation (22) is made up of known option values and the known constant coefficients  $P_u$ ,  $P_m$ ,  $P_d$  and can therefore be considered a known constant. So the set of equations (22) for  $j = -N_j + 1, \dots, N_j - 1$ , together with the boundary conditions,

$$C_{i,N_j} - C_{i,N_j-1} = \lambda_u \dots (23)$$

$$C_{i,-N_j+1} - C_{i,-N_j} = \lambda_L \dots (24)$$

make up a tri-diagonal system of equations similar to the implicit finite difference method. This system of equations can be solved very efficiently (refer to equation (13)).

The accuracy of this method is  $O(\Delta x^2 + [\Delta t/2]^2)$  and it is unconditionally stable and convergent. The Crank-Nicolson method converges much faster than the implicit or explicit finite difference methods.[3]

#### **4.COMPUTATIONAL IMPLEMENTATION/METHODOLOGY**

How these methods can be applied to the pricing of one-factor derivatives have been explained.

I will proceed to describe how the finite difference methods that have been mentioned above are implemented in C++. The source has been coded in a main C++ program. A Microsoft visual C++ compiler or a Borland compiler both with a PSDK installed is needed to run this software.

Firstly, the derivative constants will be pre-computed by declaring them as integers and doubles.

Secondly, the asset prices at maturity will be computed by defining a  $(2*N+1)$  array and putting the results in the array.

Next, the option prices at maturity will also be computed and placed in a similar array.

Finally, a  $(2*N+1)*(N)$  mesh or matrix will be created using pointers. This mesh will contain the option prices at maturity in the last column, and the same mesh will also hold discounted option prices up to today's option price in a decreasing order from N. The results will be displayed using C++ input/output stream.

**NB: Please contact Professor Domokos Vermes at [dvermes@WPLEDU](mailto:dvermes@WPLEDU) for source codes/software.**

## **5.CONCLUSION**

European call and put options ,and American call and put options were priced using C++.

The importance of this project is in its pragmability.Since it is a console factory application, a user only needs to understand how to price derivates or understand the finite difference methods in order to use this software.

It is hoped that this project will provide a foundation for other students who would wish to implement derivatives pricing using other finite difference methods.It may also serve as a source of reference.The project can be extended to price options by the Crank-Nicolson method by copying the procedure for the Implicit finite difference.

## 6.REFERENCES

[1]HULL,JOHN C. OPTIONS,FUTURES,AND OTHER DERIVATIVES.UPPER SADDLE RIVER,NEW JERSEY,U.S.A.:PEARSON EDUCATION INC.,2002,pp.152-155.

[2] [http://en.wikipedia.org/wiki/Derivative\\_\(finance\)](http://en.wikipedia.org/wiki/Derivative_(finance))

[3]CLEWLOW,LES AND STRICKLAND,CHRIS. Implementing Derivatives Models. WEST SUSSEX,ENGLAND:JOHN WILEY AND SONS LTD,THE ATRIUM,SOUTHERN GATE,CHICHESTER, July 2006,pp.57-77.